An Introduction to Sequential Rule Mining

Philippe Fournier-Viger
http://www.philippe-Fournier-viger.com

Source code and datasets available in the SPMF library

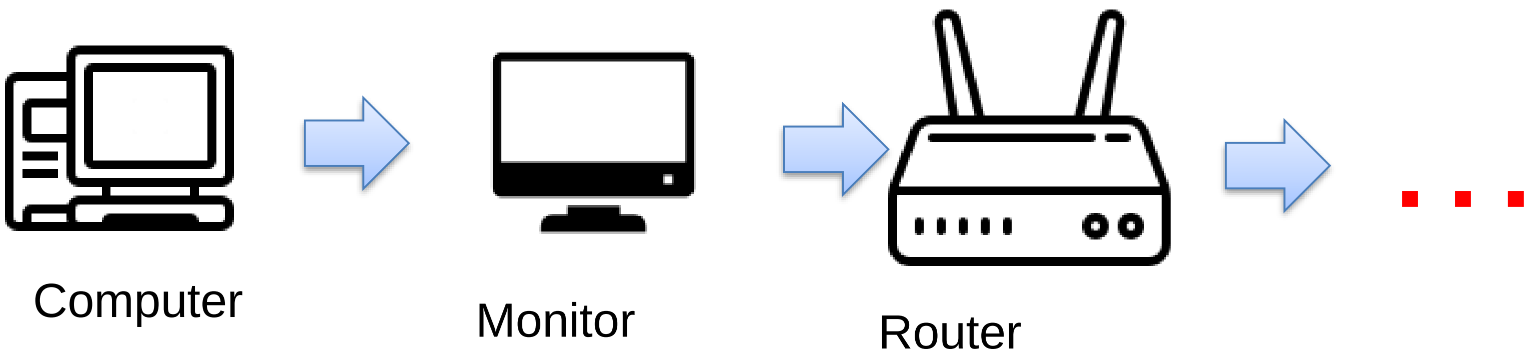
Introduction

- More and more data!
- A need to analyze data to find **interesting patterns**
- **Pattern mining**: using algorithms to find interesting patterns in data.
- An important type of data is **sequences**.
- Today, we will discuss how to analyze sequences to find **sequential rules**.

What is a **discrete sequence**?

Sequence: an ordered list of symbols

Sequence of purchases



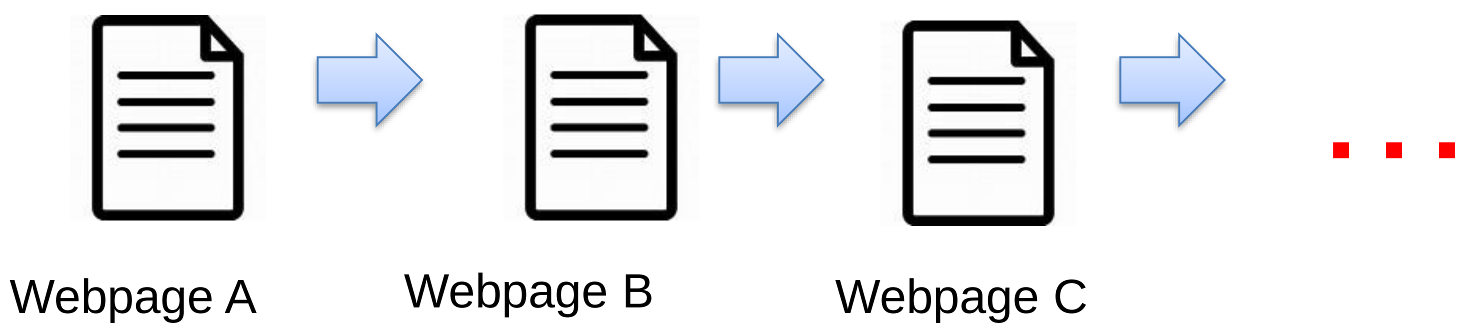
Sequence of words

Where → **are** → **you** → **going?**

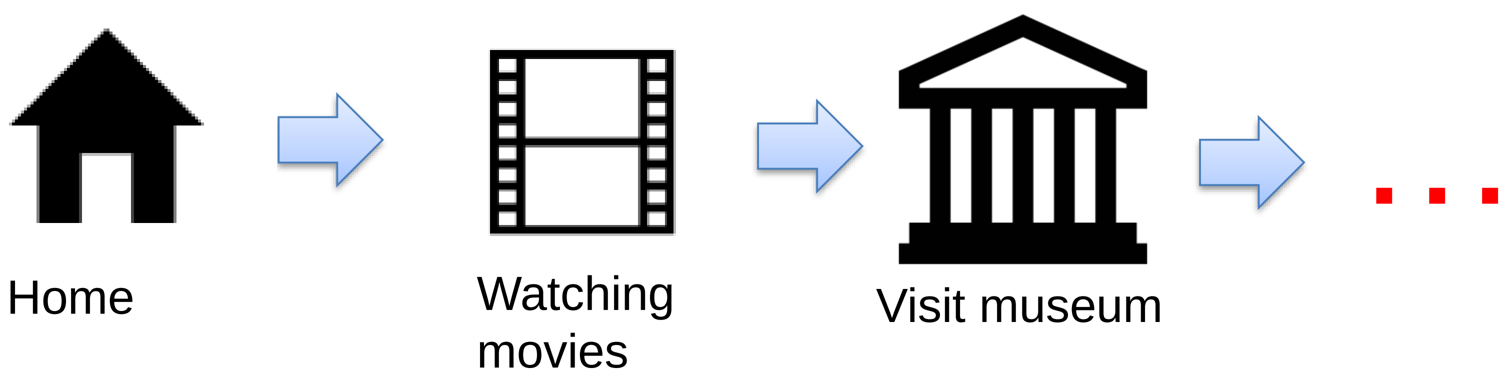
What is a **discrete sequence**?

Sequence: an ordered list of symbols

Sequences of webpage clicks



Sequences of activities

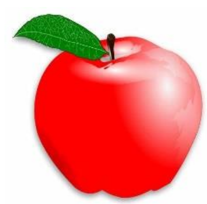


Definition: Items

Let there be a **set** of **items** (symbols) called I.

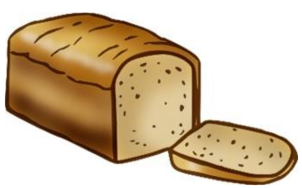
Example: $I = \{a, b, c, d, e, f, g\}$

a = apple

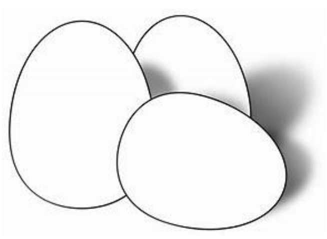


d = dattes

b = bread

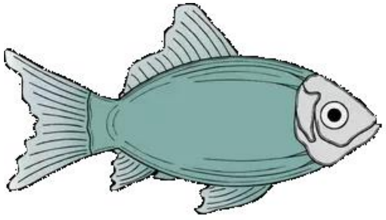


e = eggs



c = cake

f = fish

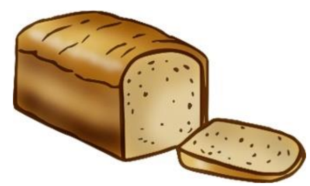


g = grapes

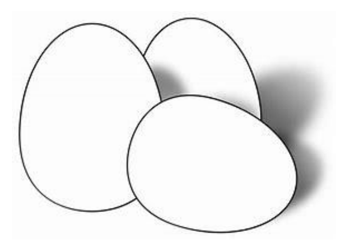
Definition: Itemset

An itemset is a set of **items** that is a subset of I.

Example: $\{a, b, c\}$ is an itemset containing 3 items


$\{d, e\}$ is an itemset containing 2 items

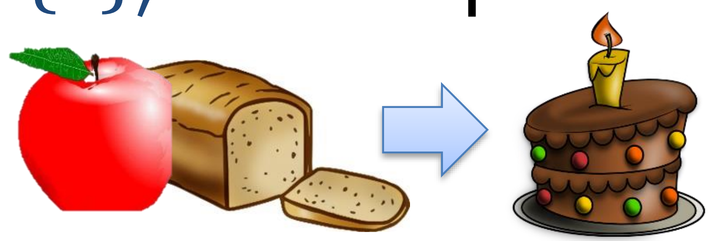


- Note: an itemset cannot contain a same item twice.
- An itemset having k items is called a *k-itemset*.

Definition: Sequence

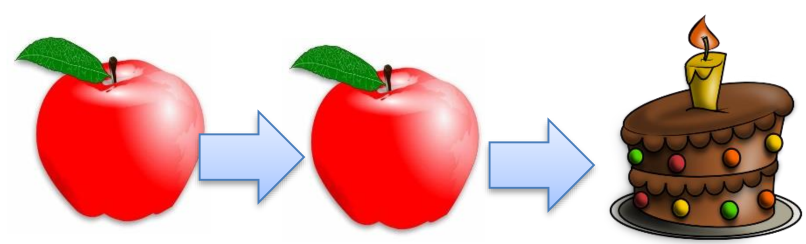
A **discrete sequence** S is a an ordered list of itemsets $S = \langle X_1, X_2, \ldots, X_n \rangle$ where $X_j \subseteq I$ for any $j \in \{1,2..n\}$

Example 1: $\langle \{a, b\}, \{c\} \rangle$ is a sequence containing two itemsets.



It means that a customer purchased $apple$ and $bread$ at the same time and then purchased $cake$.

Example 2: $\langle \{a\}, \{a\}, \{c\} \rangle$



Definition: Sequence Database

- A **sequence database** is one or more sequences.

SID	sequence
1	<{a}, {a,b,c} {a, c} {d} {c, f}>
2	<{a, d}, {c} {b, c} {a, e}>
3	<{e, f}, {a, b} {d, f} {c}, {b}>
4	<{e}, {g}, {a, f} {c} {b}, {c}>

- Here we have four sequences.
- Each sequence has a unique sequence identifier (SID)

Sequential pattern mining

It is a popular data mining task, where the goal is to find **sequential patterns.**

SID	sequence
1	<{a}, {a,b,c} {a, c} {d} {c, f}>
2	<{a, d}, {c} {b, c} {a, e}>
3	<{e, f}, {a, b} {d, f} {c}, {b}>
4	<{e}, {g}, {a, f} {c} {b}, {c}>

Sequential pattern mining

Sequential pattern: a subsequence that appear in many sequences of a sequence database

SID	sequence
1	<{a}, {a,b,c} {a, c} {d} {c, f}>
2	<{a, d}, {c} {b, c} {a, e}>
3	<{e, f}, {a, b} {d, f} {c}, {b}>
4	<{e}, {g}, {a, f} {c} {b}, {c}>

<{a},{f}> is a **sequential pattern**

Sequential pattern mining

Sequential pattern: a subsequence that appear in many sequences of a sequence database

SID	sequence
1	<{a}, {a,b,c} {a, c} {d} {c, f}>
2	<{a, d}, {c} {b, c} {a, e}>
3	<{e, f}, {a, b} {d, f} {c}, {b}>
4	<{e}, {g}, {a, f} {c} {b}, {c}>

<{a},{f}> is a **sequential pattern**
Its **support** is 50% (it appears in 50% of the sequences).

Sequential pattern mining

Input:

- A sequence database (a set of sequences)
- A ***minsup*** threshold

Output:

- All subsequences having a support greater or equal to ***minsup***.

Example: minsup = 50 % (2 sequences)

A sequence database

IFD	sequence
1	<{a}, {a,b,c} {a, c} {d} {c, f}>
2	<{a, d}, {c} {b, c} {a, e}>
3	<{e, f}, {a, b} {d, f} {c}, {b}>
4	<{e}, {g}, {a, f} {c} {b}, {c}>

→

Sequential patterns

Pattern	support
{a}	100 %
<{a}, {b,c} >	50 %
<{a, b} >	50 %
…	…

Fournier-Viger, P., Lin, J. C.-W., Kiran, R. U., Koh, Y. S., Thomas, R. (2017). A Survey of Sequential Pattern Mining. Data Science and Pattern Recognition (DSPR), vol. 1(1), pp. 54-77.

Some popular algorithms

- **GSP**: R. Agrawal, and R. Srikant, Mining sequential patterns, ICDE 1995, pp. 3–14, 1995.
- **SPAM:** Ayres, J. Flannick, J. Gehrke, and T. Yiu, Sequential pattern mining using a bitmap representation, KDD 2002, pp. 429–435, 2002.
- **SPADE**: M. J. Zaki, SPADE: An efficient algorithm for mining frequent sequences, Machine learning, vol. 42(1-2), pp. 31–60, 2001.
- **PrefixSpan**: J. Pei, et al. Mining sequential patterns by pattern-growth: The prefixspan approach, IEEE Transactions on knowledge and data engineering, vol. 16(11), pp. 1424–1440, 2004.
- **CM-SPAM** and **CM-SPADE**: P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas, Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information, PAKDD 2014, pp. 40–52, 2014.

They all have the same input and output.
The difference is performance due to optimizations, search strategies and data structures!

Fast implementations available in the SPMF library

But there is a problem…

Let look at the pattern **<{a},{f}>**

We might think that if someone buys « a », he will he buy « f » afterward.

SID	sequence
1	<{**a**}, {a,b,c} {a, c} {d} {c, **f**}>
2	<{a, d}, {c} {b, c} {a, e}>
3	<{e, f}, {**a**, b} {d, **f**} {c}, {b}>
4	<{e}, {g}, {a, f} {c} {b}, {c}>

But there is a problem…

Let look at the pattern **<{a},{f}>**

We might think that if someone buys « a », he will he buy « f » afterward.

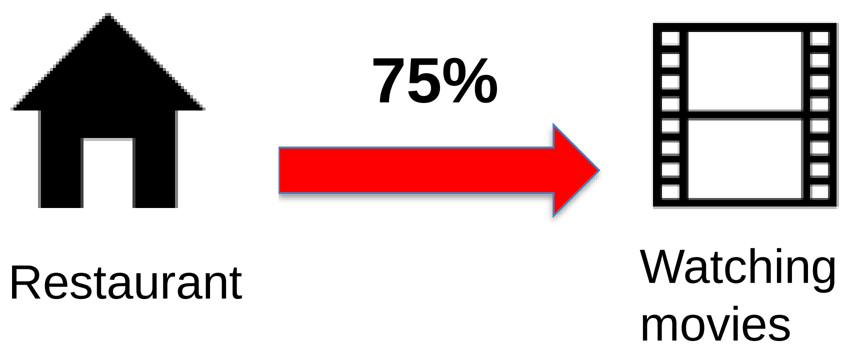
No ! Only 50% of the time!

SID	sequence
1	<{**a**}, {a,b,c} {a, c} {d} {c, **f**}>
2	<{a, d}, {c} {b, c} {a, e}>
3	<{e, f}, {**a**, b} {d, **f**} {c}, {b}>
4	<{e}, {g}, {a, f} {c} {b}, {c}>

Thus, sequential patterns can be **misleading**!

How to address this problem?

- We would like to find patterns that have the form of rules.
- We want to measure the confidence (probability) that some item(s) will follow some other item(s).
- **Solution:** finding **sequential rules**



Two main types of sequential rules

1) **Standard Sequential rules**
2) **Partially-ordered Sequential rules**

1) Standard Sequential rules

Standard Sequential rules: Rules of the form $X \rightarrow Y$, where X and Y are sequential patterns.

Example: <{a}, {b,c}> → <{d}, {e}>

1) Standard Sequential rules

Standard Sequential rules: Rules of the form $X \rightarrow Y$, where X and Y are sequential patterns.

Example: <{a}, {b,c}> → <{d}, {e}>

- Several algorithms to find this type of rules such as **RuleGen** (Zaki,2001).
- **Main idea**: find sequential patterns and then combine them to make rules.

1) Standard Sequential rules

Standard Sequential rules: Rules of the form $X \rightarrow Y$, where X and Y are sequential patterns.

Example: <{a}, {b,c}> → <{d}, {e}>

- Two thresholds must be set by the user:
  - minimum support > 0
  - minimum confidence > 0
- **Support**: how many sequences contain a rule
- **Confidence**: how many sequences contain a rule divided by how many sequences contain its antecedent

1) Standard Sequential rules

Example*:* <{a}, {b}> → <{f}>

Support: 1 sequences (25%)
Confidence: 1 / 4 = 0.25 (25%)

SID	sequence
1	<{a}, {a,b,c} {a, c} {d} {c, f}>
2	<{a, d}, {c} {b, c} {a, e}>
3	<{e, f}, {a, b} {d, f} {c}, {b}>
4	<{e}, {g}, {a, f} {c} {b}, {c}>

But there is a problem…

We may find some sequential rules that are very similar but have only some **small ordering variations**.

For example:

Rule	Support	Confidence
<{a}, {b}> → <{f}>	25%	25%
<{b}, {a}> → <{f}>	25%	50%

These rules may actually represent the same situation!

2) Partially-Ordered Sequential rules

Partially-Ordered Sequential rules: Rules of the form $X \rightarrow Y$, where X and Y are itemsets that are **unordered**.

Example: {a,b} → {f}

2) Partially-Ordered Sequential rules

Partially-Ordered Sequential rules: Rules of the form $X \rightarrow Y$, where X and Y are itemsets that are **unordered**.

Example: {a,b} → {f}

Interpretation: If we observe **a** and **f** (in any order), they will be followed by **f**.

2) Partially-Ordered Sequential rules

- This type of rule is often more interesting because it can summarize many standard sequential rules.
- **For example:**

$\{Vivaldi\}, \{Mozart\}, \{Handel\} \Rightarrow \{Berlioz$
$\{Mozart\}, \{Vivaldi\}, \{Handel\} \Rightarrow \{Berlioz\},$
$\{Handel\}, \{Vivaldi\}, \{Mozart\} \Rightarrow \{Berlioz\},$
$\{Handel, Vivaldi\}, \{Mozart\} \Rightarrow \{Berlioz\},$
$\{Handel\}, \{Vivaldi, Mozart\} \Rightarrow \{Berlioz\},$
$\{Handel, Vivaldi, Mozart\} \Rightarrow \{Berlioz\}.$

Standard sequential rules

$\{Mozart, Vivaldi, Handel\} \Rightarrow \{Berlioz\}$

Partially-ordered sequential rules

2) Partially-Ordered Sequential rules

- A **partially-ordered sequential rule** $X \rightarrow Y$ is a relationship between two disjoint and non empty itemsets X,Y.
- A sequential rule $X \rightarrow Y$ has **two properties**:
  - **Support:** the number of sequences where X occurs before Y, divided by the number of sequences.
  - **Confidence** the number of sequences where X occurs before Y, divided by the number of sequences where X occurs.
- **The task**: finding all **valid rules**, rules with a support and confidence not less than user-defined thresholds *minSup* and *minConf* (Fournier-Viger, 2010).

An example of Sequential Rule Mining

Let say that *minSup*= 0.5 and *minConf*= 0.5:

ID	Sequences
seq1	{*a*, *b*},{*c*},{*f*},{*g*},{*e*}
seq2	{*a*, *d*},{*c*},{*b*},{*a*, *b*, *e*, *f*}
seq3	{*a*},{*b*},{*f*},{*e*}
seq4	{*b*},{*f*, *g*}

A sequence database

→

ID	Rule	Support	Confidence
r1	{*a*, *b*, *c*}⇒{*e*}	0.5	1.0
r2	{*a*}→{*c*, *e*, *f*}	0.5	0.66
r3	{*a*, *b*} →{*e*, *f*}	0.75	1.0
r4	{*b*} →{*e*, *f*}	0.75	0.75
r5	{*a*} →{*e*, *f*}	0.75	1.0
r6	{*c*} →{*f*}	0.5	1.0
r7	{*a*}→{*b*}	0.5	0.66
…	…	…	…

Some rules found

Several algorithms

- **CMRules, RuleGrowth, ERMiner**: find all the sequential rules
- **TRuleGrowth**: find sequential rules with a window constraint
- **TopSeqRules**: find the top-k sequential rules
- **TNS**: find top-k non-redundant sequential rules
- **HUSRM**: find high utility sequential rules
- …

These algorithms directly find the rules!

Some applications

E-learning

- Fournier-Viger, P., Faghihi, U., Nkambou, R., Mephu Nguifo, E.: CMRules: Mining Sequential Rules Common to Several Sequences. Knowledge-based Systems, Elsevier, 25(1): 63-76 (2012)
- Toussaint, Ben-Manson, and Vanda Luengo. "Mining surgery phase-related sequential rules from vertebroplasty simulations traces." Artificial Intelligence in Medicine. Springer International Publishing, 2015. 35-46.
- Faghihi, Usef, Philippe Fournier-Viger, and Roger Nkambou. "CELTS: A Cognitive Tutoring Agent with Human-Like Learning Capabilities and Emotions." Intelligent and Adaptive Educational-Learning Systems. Springer Berlin Heidelberg, 2013. 339-365.

Some applications

Manufacturing simulation

- Kamsu-Foguem, B., Rigal, F., Mauget, F.: Mining association rules for the quality
improvement of the production process. Expert Systems and Applications 40(4),
1034-1045 (2012)

Quality control

- Bogon, T., Timm, I. J., Lattner, A. D., Paraskevopoulos, D., Jessen, U., Schmitz,
M., Wenzel, S., Spieckermann, S.: Towards Assisted Input and Output Data Analysis
in Manufacturing Simulation: The EDASIM Approach. In: Proc. 2012 Winter
Simulation Conference, pp. 257–269 (2012)

Some applications

Web page prefetching

- Fournier-Viger, P. Gueniche, T., Tseng, V.S.: Using Partially-Ordered Sequential Rules to Generate More Accurate Sequence Prediction. Proc. 8th International Conference
on Advanced Data Mining and Applications, pp. 431-442, Springer (2012)

Anti-pattern detection in service based systems,

- Nayrolles, M., Moha, N., Valtchev, P.: Improving SOA antipatterns detection in Service Based Systems by mining execution traces. In: Proc. 20th IEEE Working Conference on Reverse Engineering, pp. 321-330 (2013)

Embedded systems

- Leneve, O., Berges, M., Noh, H. Y.: Exploring Sequential and Association Rule Mining for Pattern-based Energy Demand Characterization. In: Proc. 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings. ACM, pp. 1–2 (2013)

Some applications

Alarm sequence analysis

- Celebi, O.F., Zeydan, E., Ari, I., Ileri, O., Ergut, S.: Alarm Sequence Rule Mining Extended With A Time Confidence Parameter. In: Proc. 14th Industrial Conference on Data Mining (2014)
- Ileri, Omer, and Salih Ergüt. "Alarm Sequence Rule Mining Extended With A Time Confidence Parameter." (2014).

Recommendation

- Jannach, Dietmar, and Simon Fischer. "Recommendation-based modeling support for data mining processes." Proceedings of the 8th ACM Conference on Recommender systems. ACM, 2014.

Some applications

Restaurant recommendation

- Han, M., Wang, Z., Yuan, J.: Mining Constraint Based Sequential Patterns and Rules on Restaurant Recommendation System. Journal of Computational Information Systems 9(10), 3901-3908 (2013)

Customer behavior analysis

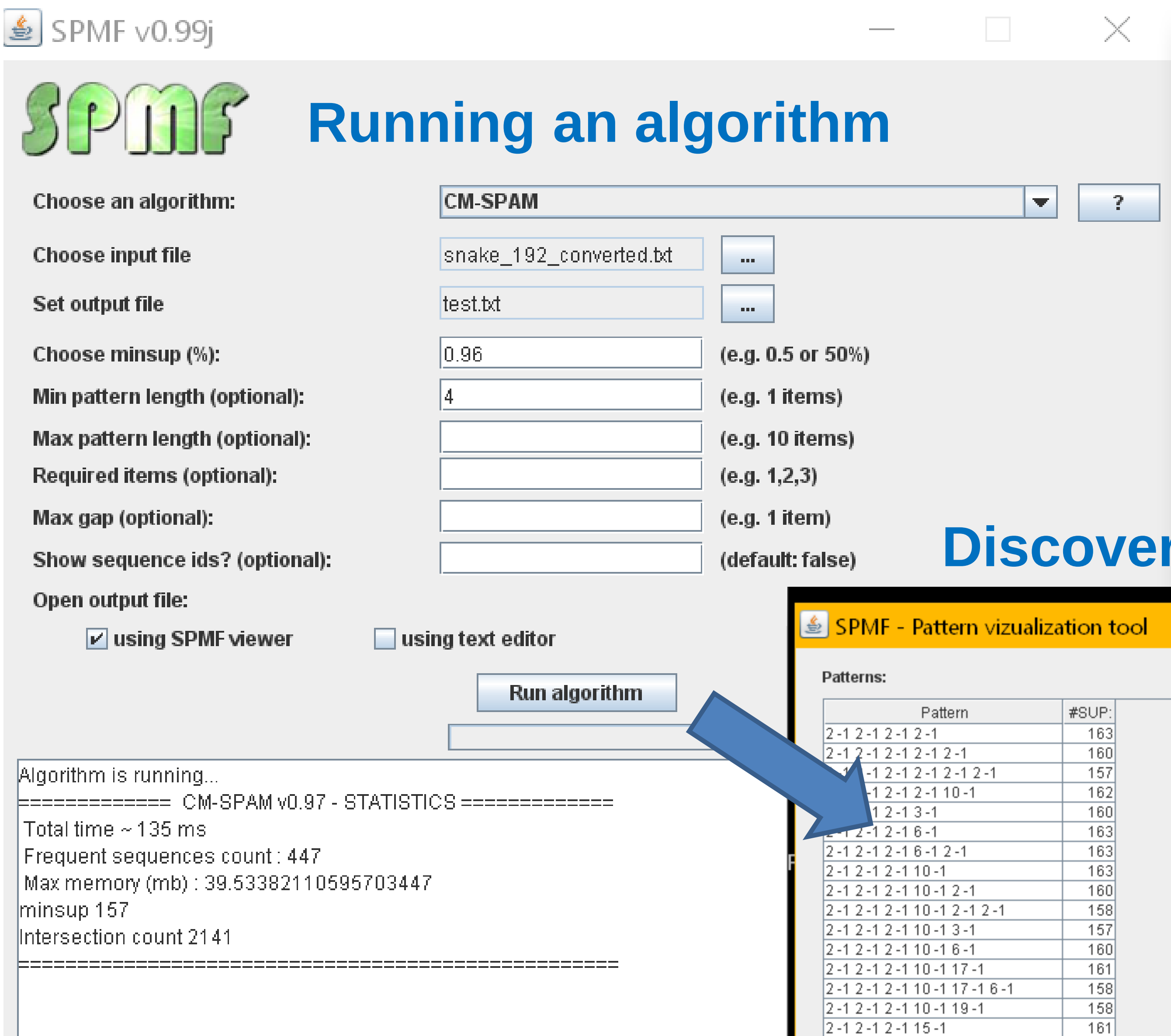
- Noughabi, Elham Akhond Zadeh, Amir Albadvi, and Behrouz Homayoun Far. "How Can We Explore Patterns of Customer Segments' Structural Changes? A Sequential Rule Mining Approach." Information Reuse and Integration (IRI), 2015 IEEE International Conference on. IEEE, 2015.

Conclusion

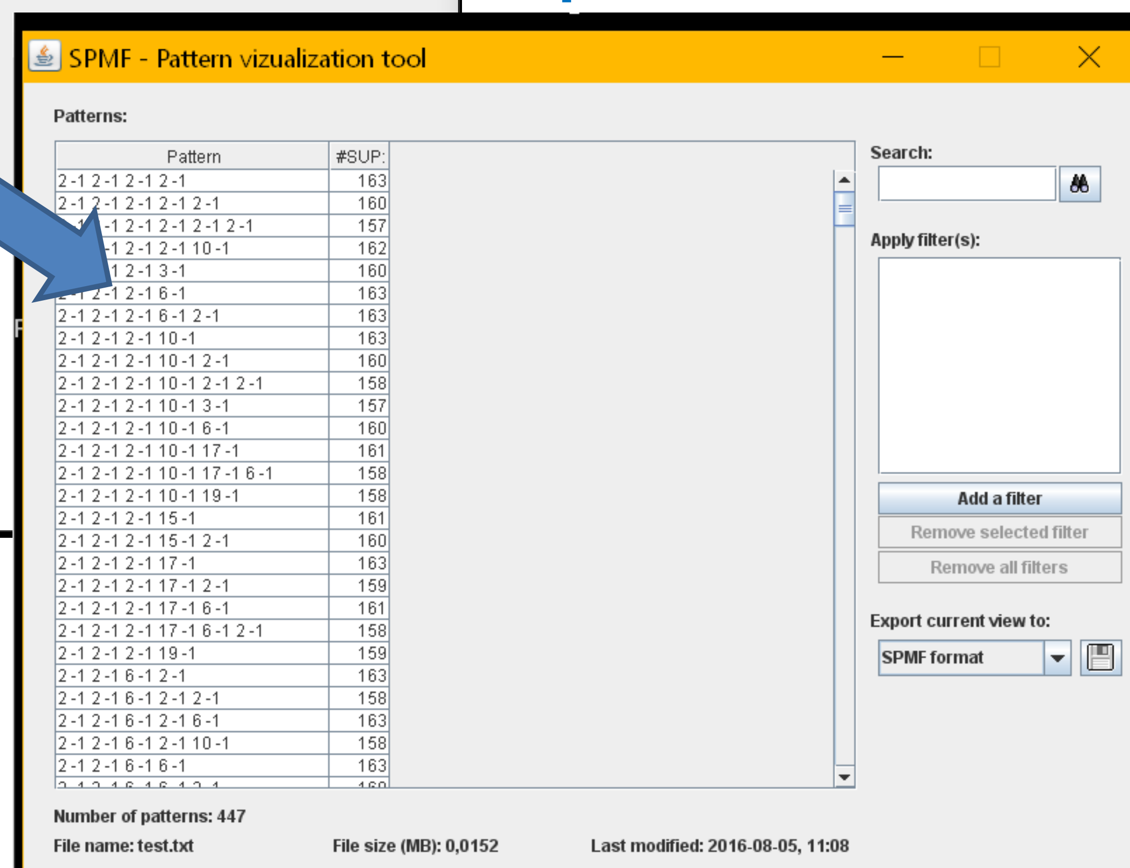
- Today, I have introduced **sequential rule mining**.
- An important topic in pattern mining.
- Sometimes also called **temporal association rule mining** or **episode rules**.
- There are also other variations.

- Source code and dataset in the **SPMF library**

Running an algorithm



Discovered patterns



http://www.philippe-fournier-viger.com/spmf/